Development of Automation system for Network acceptance test and Prospect for operation process improvement

Okinawa Open Laboratory
Network Test system Project

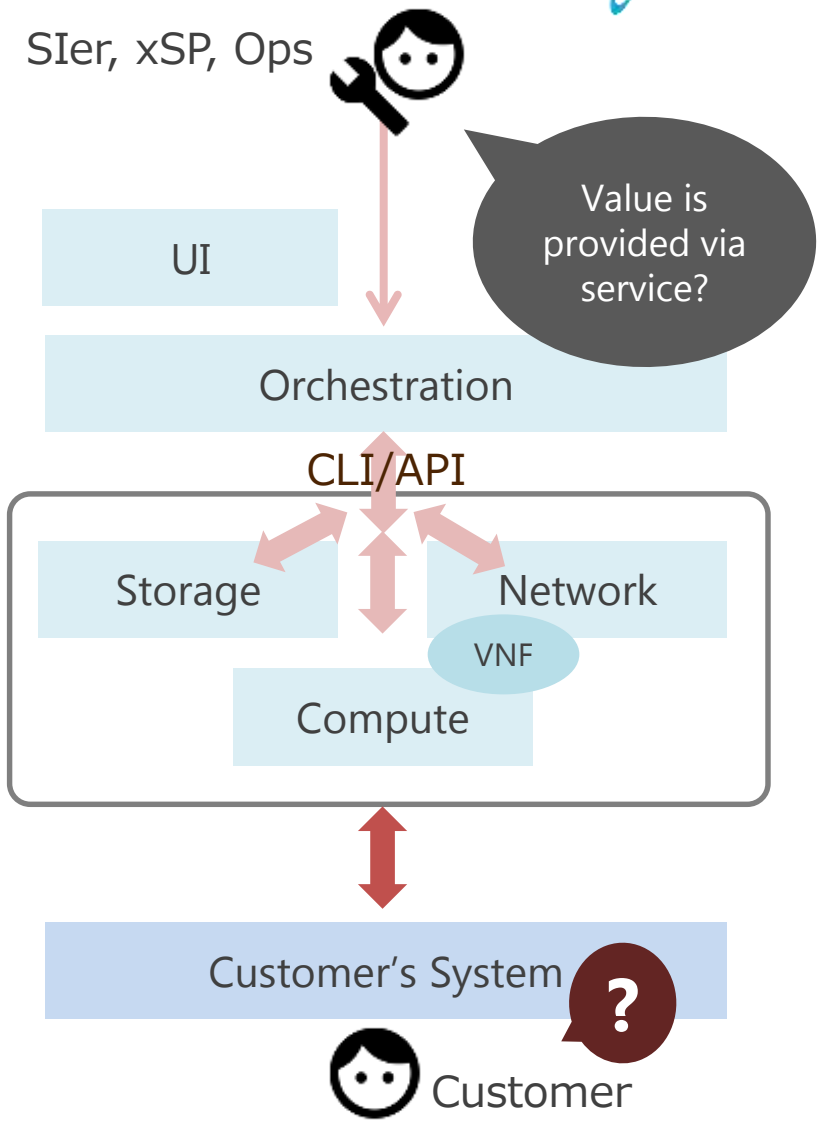
Introduction

Configure and Operate Network

- Configure system and services by connecting a bunch of various devices to operate via CLI/API

Making Decision for service is important

- Combination of many black boxes
- After configuration devices, How they behaves as system?
- Verify the value is provided for customer properly along to user's business cycle

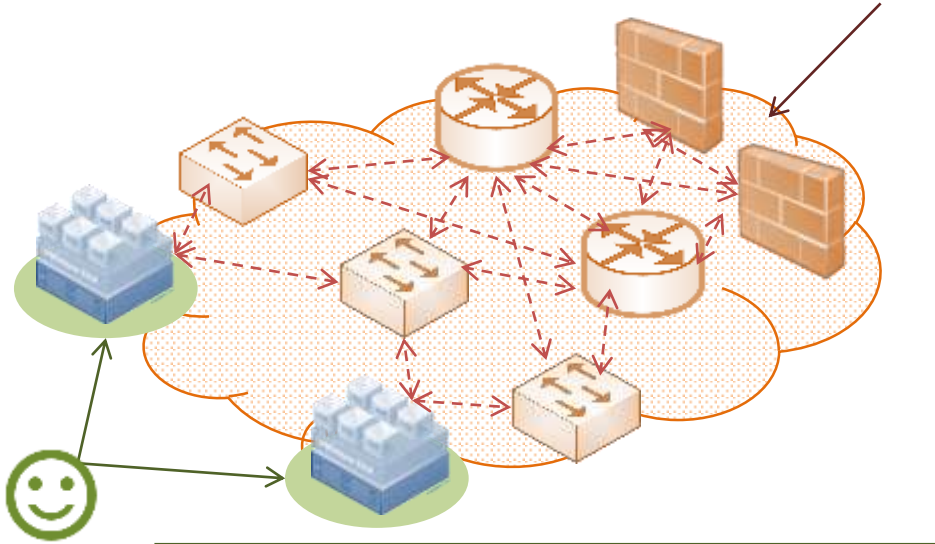


Problem on Network

Difficult to automate

- Difference of policies
  - Put units sequentially, and expected to connect naturally
  - No guarantee to connect, Guarantee that was connected
    - Interoperability check for many devices
- Adjust on site If not connected
  - On site, On device, Manpower
  - Human wave tactics...
- Different operation interface per each devices

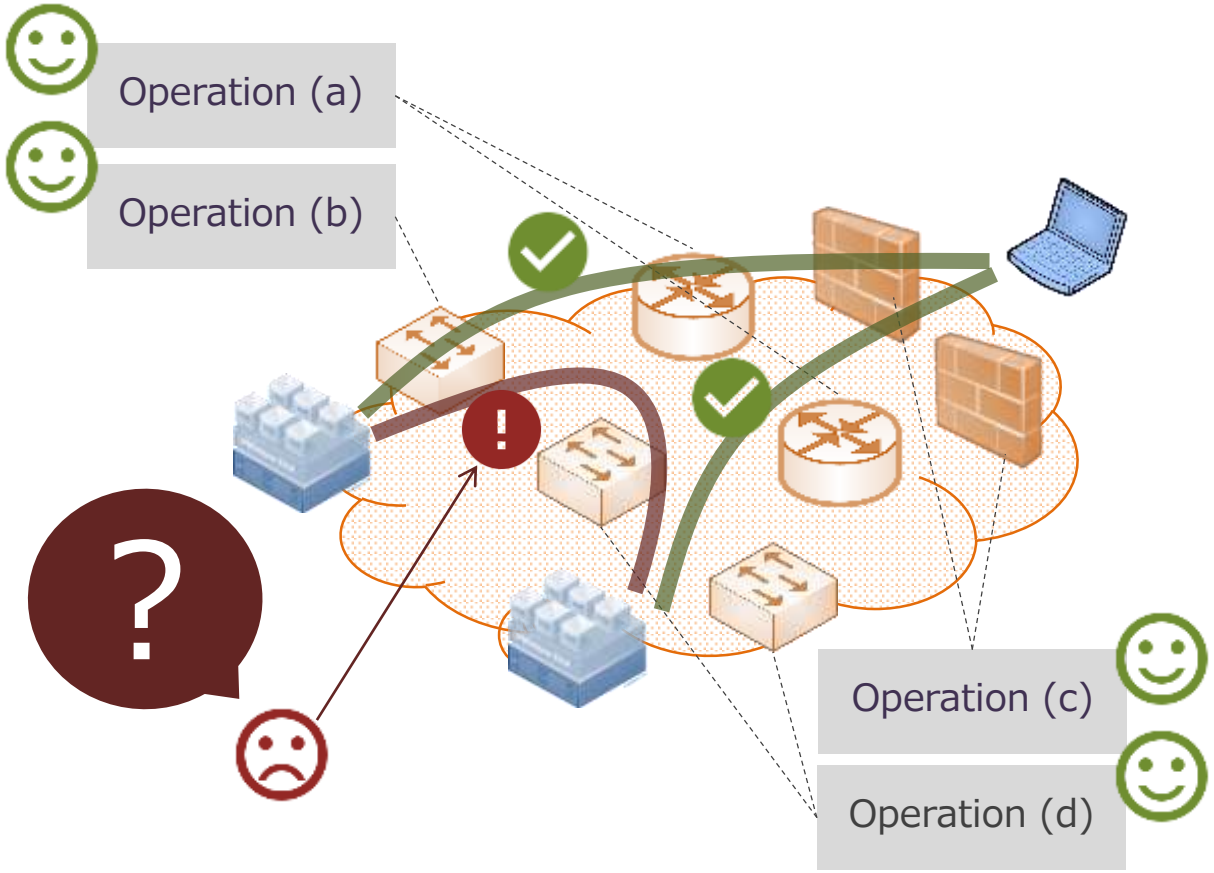
“Network made manually”
Confirm interoperability for various devices...
Test? Manual test on site...



”Network is connected naturally”
Once connected. Configuration is simple
Focus on a test with various tool

Difficult to confirm total operation

- **Is it expected operation or status as defined?**
  - Success for each operations ≠ Success Total operation
- Difference between devices → Many test cases
  - Difficult to confirm affection range



Result…

- Manual work, operation
- Difficult to predict affection range, review procedures, again and again...
- Lack was found on site...
  - Cost is necessary to confirm in verification environment
  - It’s too difficult to cover all of cases

NW is a bottleneck of agility for system (service)

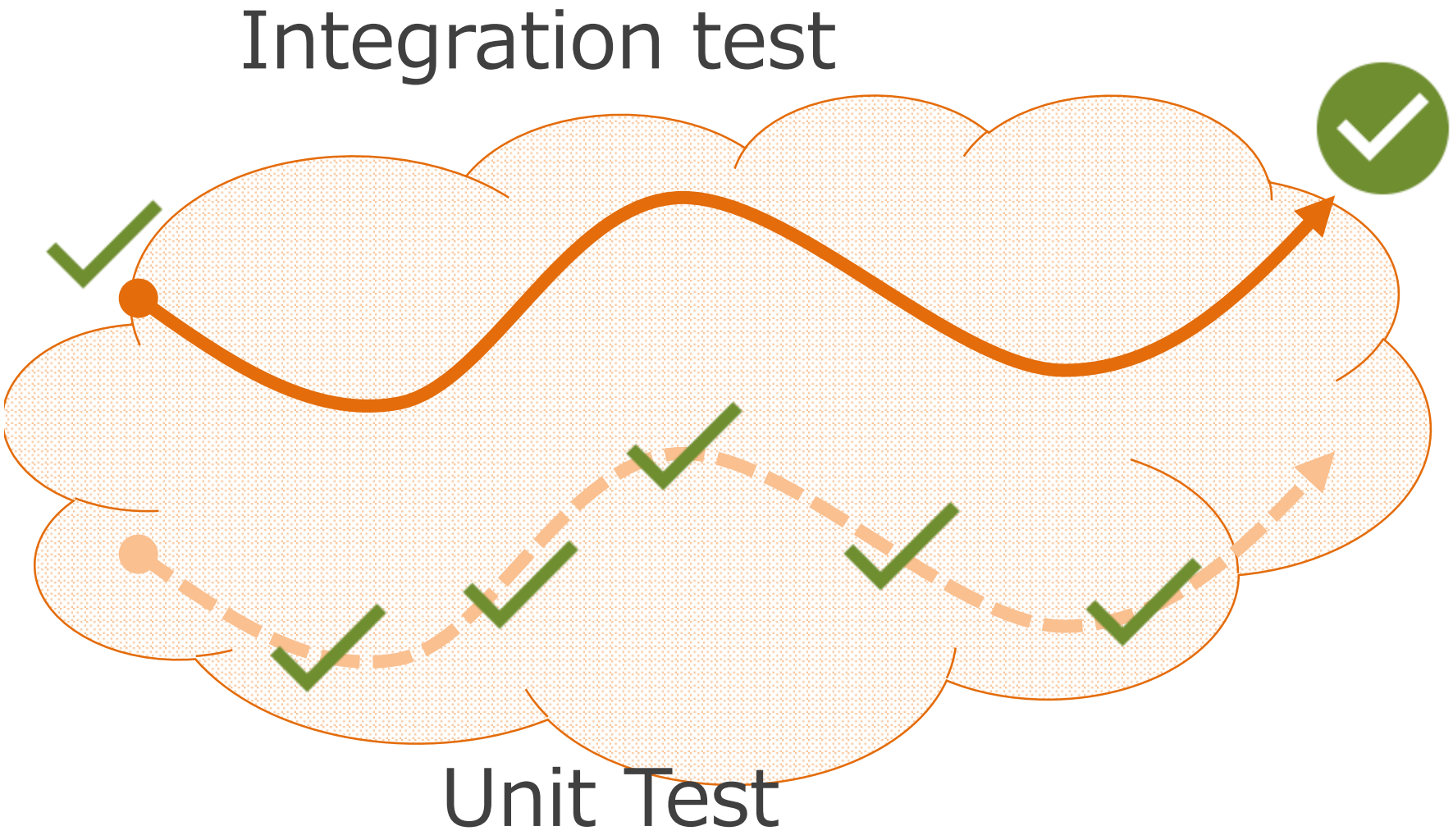
Approach

Approach to solve

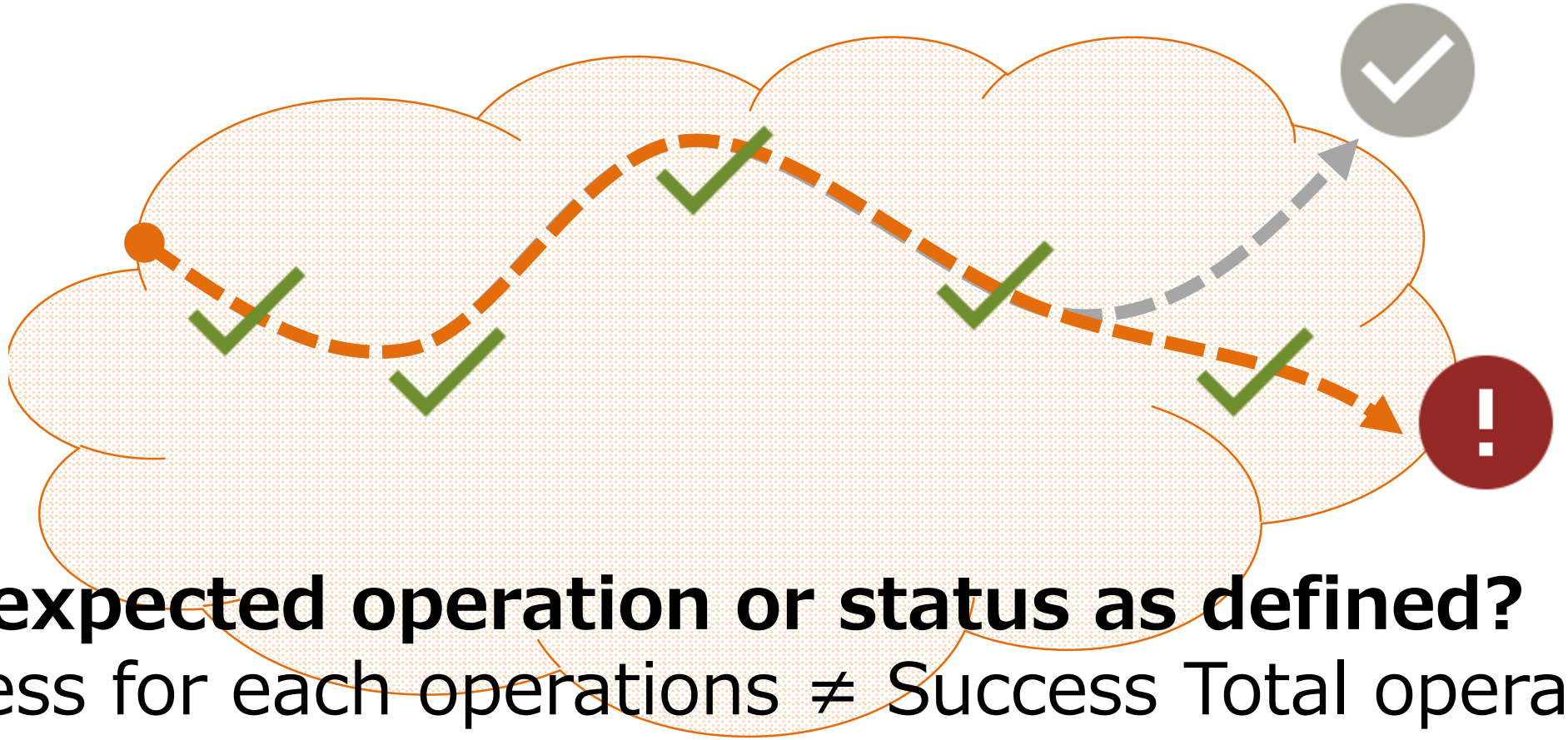
- There is a limitation to cover large scale and complicated system with Human power
  - **Test system In-Service or In-use**
  - Test large number of cases as quick as impossible for human
    - Test Automation

- Approach: “BDD” and Reason why
- Behavior of network and usecase of test
- Necessary functions for automation and target of FY2016

About BDD



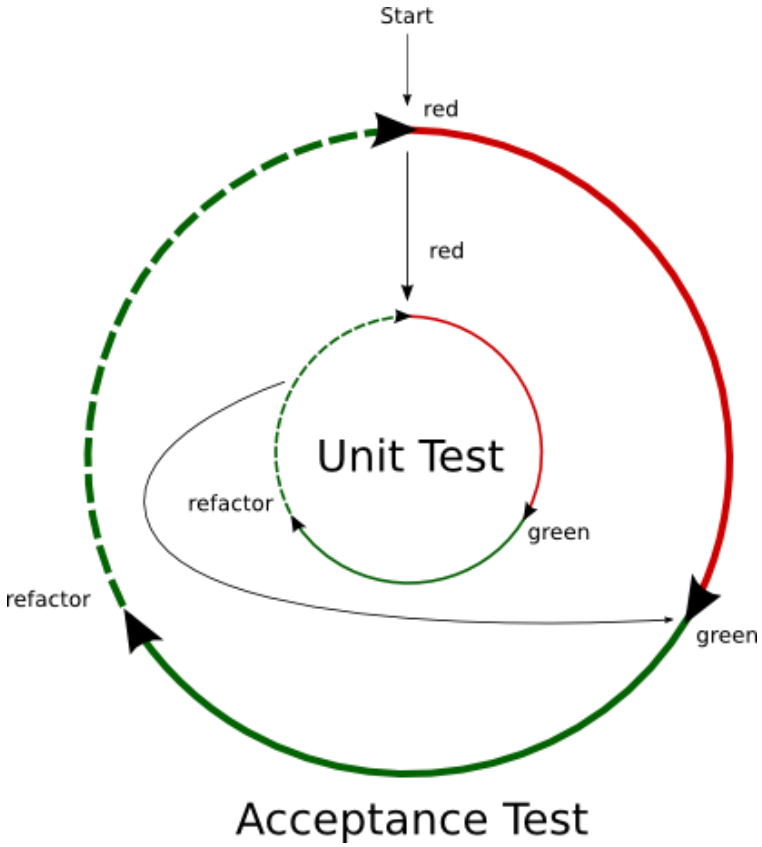
Why BDD?



Is it expected operation or status as defined?
Success for each operations ≠ Success Total operation

Why BDD?

- **Figure out purpose of test**
  - Expectation is how it behaves as system/service
  - What is a specification to satisfy finally?
- **Practical test**
  - Scenario based on practical use
- **Prevent unnecessary test**
  - Define connection between upper INPUT (Specification) and unit test
  - Possible to omit detailed test if it works End-to-End



Should TDD and BDD be used in conjunction? - Stack Overow
http://stackoverflow.com/questions/33746804/should-tdd-and-bdd-be-used-in-conjunction

“Behavior” of Network?

OKINAWA OPEN LABORATORY

Test for static behavior

Communication service is provided under **stable state network**

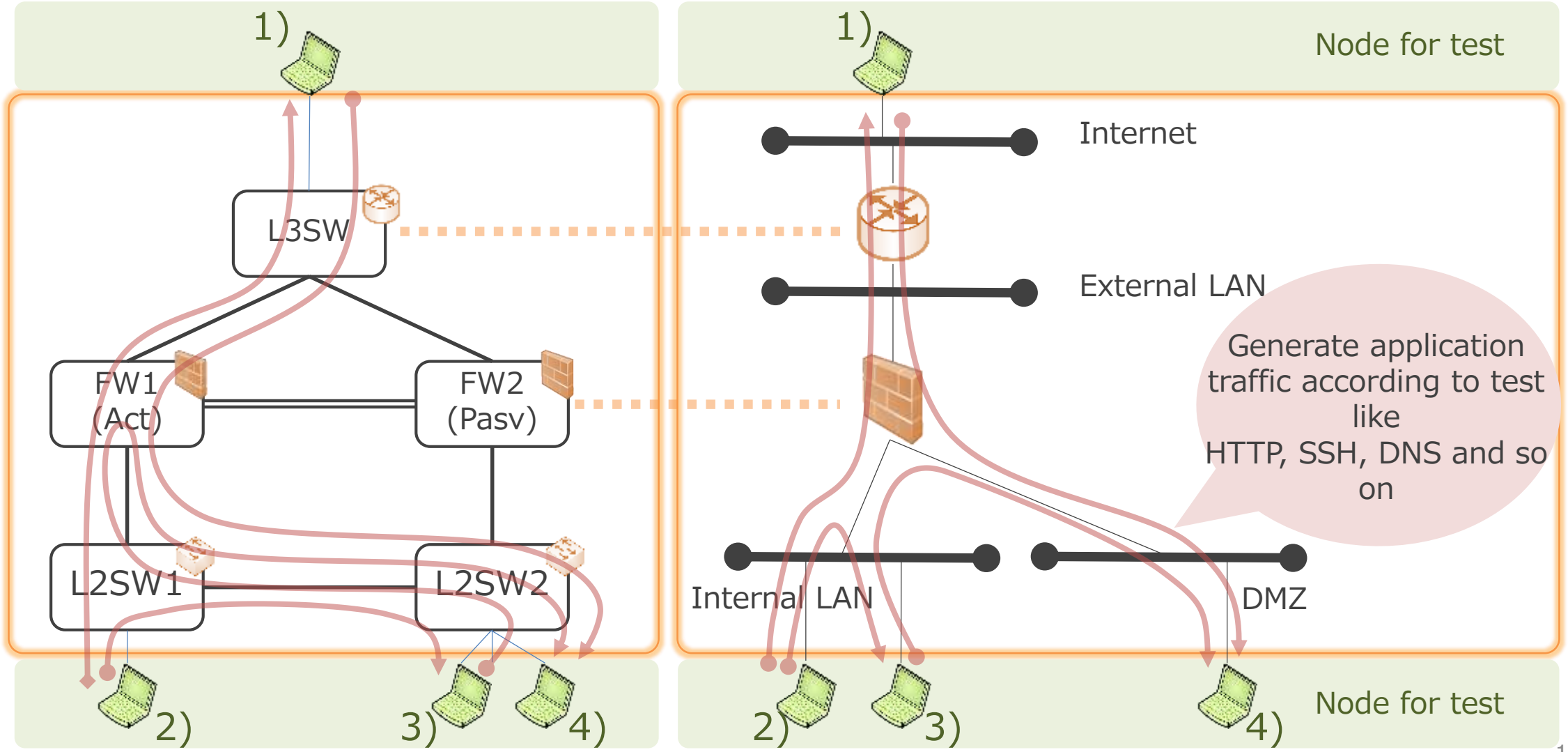
- Possible to communicate on NW…**Functional Test**

Test for dynamic behavior

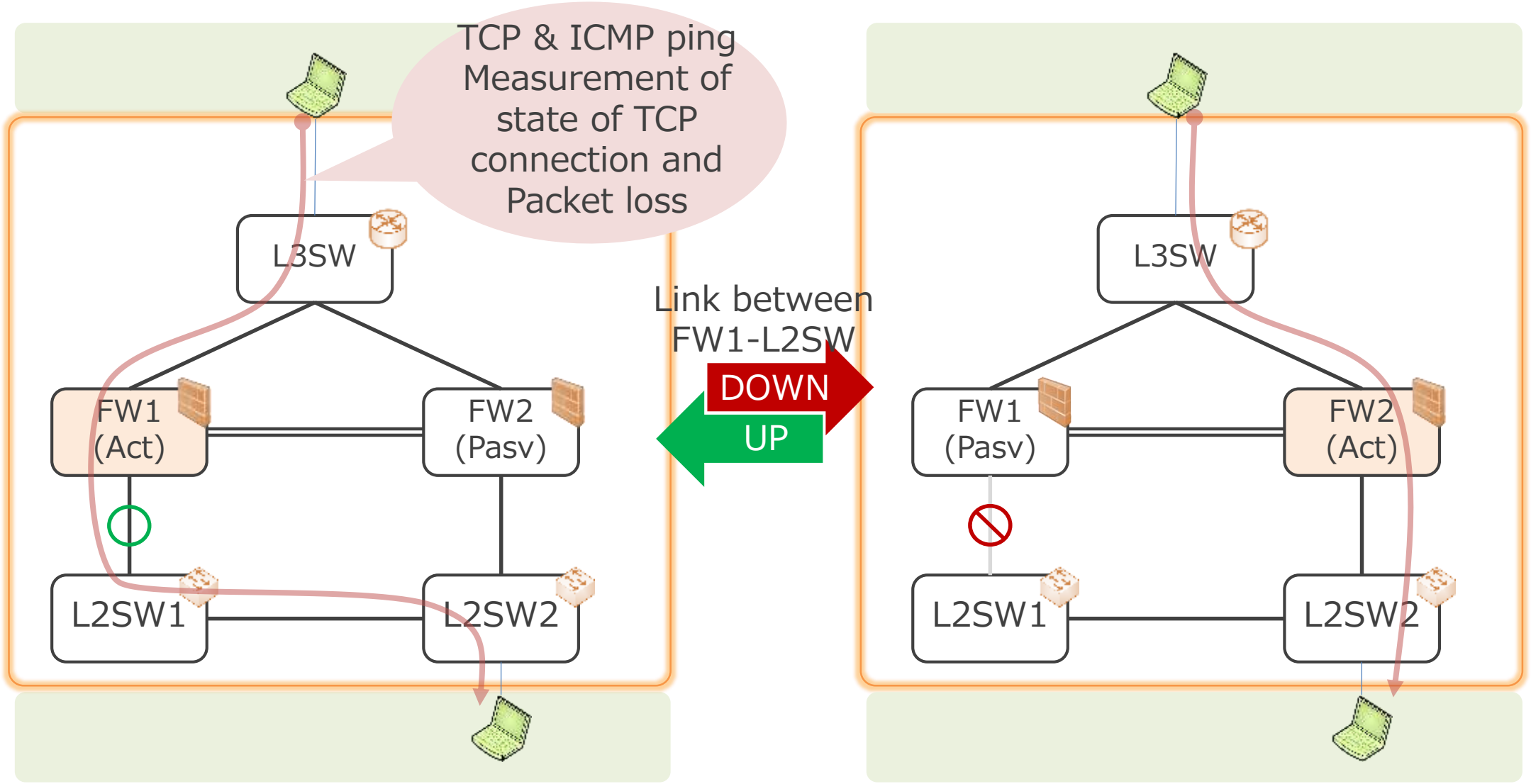
Communication service is possible to evaluate affection under **changing state network**

- Changing state of network…**failure test** (Linkdown)
- Measure on changing state timing
- How does “Communication Service” behave before and after changing state of Network

Static Test


1)
L3SW
FW1
(Act)
FW2
(Pasv)
L2SW1
L2SW2
2)
3)
4)
1)
Node for test
Internet
External LAN
Generate application traffic according to test like
HTTP, SSH, DNS and so on
Internal LAN
DMZ
2)
3)
4)
Node for test

Dynamic Test (Link failure test)



Elements of NW test automation

Create and operate NW (Topology) for Test target

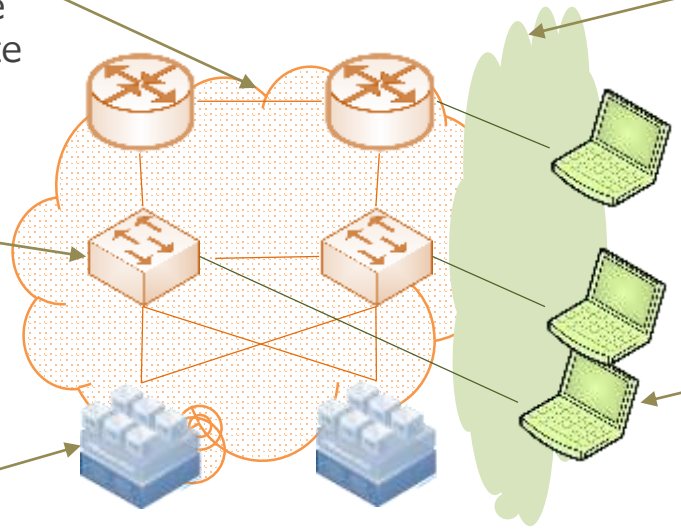
Operation for physical topology by software
Emulate failure, switch test of physical route

Configure and operate NW devices for Test target

Device configuration and retrieve state via interface of devices (CLI/REST/Netconf...)

Configure and operate Server Resources for Test target

Configure and setup physical/logical resources and services



Deploy(Connect) test node to test target

Cover test cases for physical/logical configuration
Mechanism to input/output test traffic on demand

Generate and operate test node (Execution of each test)

Generate various traffic, Tx/Rx Traffic, Judge result and simultaneous control for multi nodes (client/server etc.)

Elements of NW test automation

Create and operate NW (Topology) for Test target

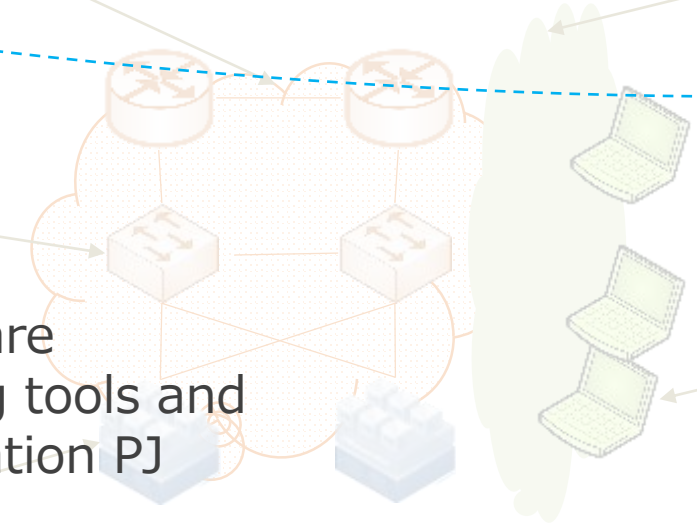
PoC with SDN was conducted on FY2015

Deploy(Connect) test node to test target

Configure and operate NW devices for Test target

There are Existing tools and automation PJ

Configure and operate Server Resources for Test target



Generate and operate test node (Execution of each test)

Target on FY2016

How can it test for detailed use case?

Architecture of Test Automation (NetTester)

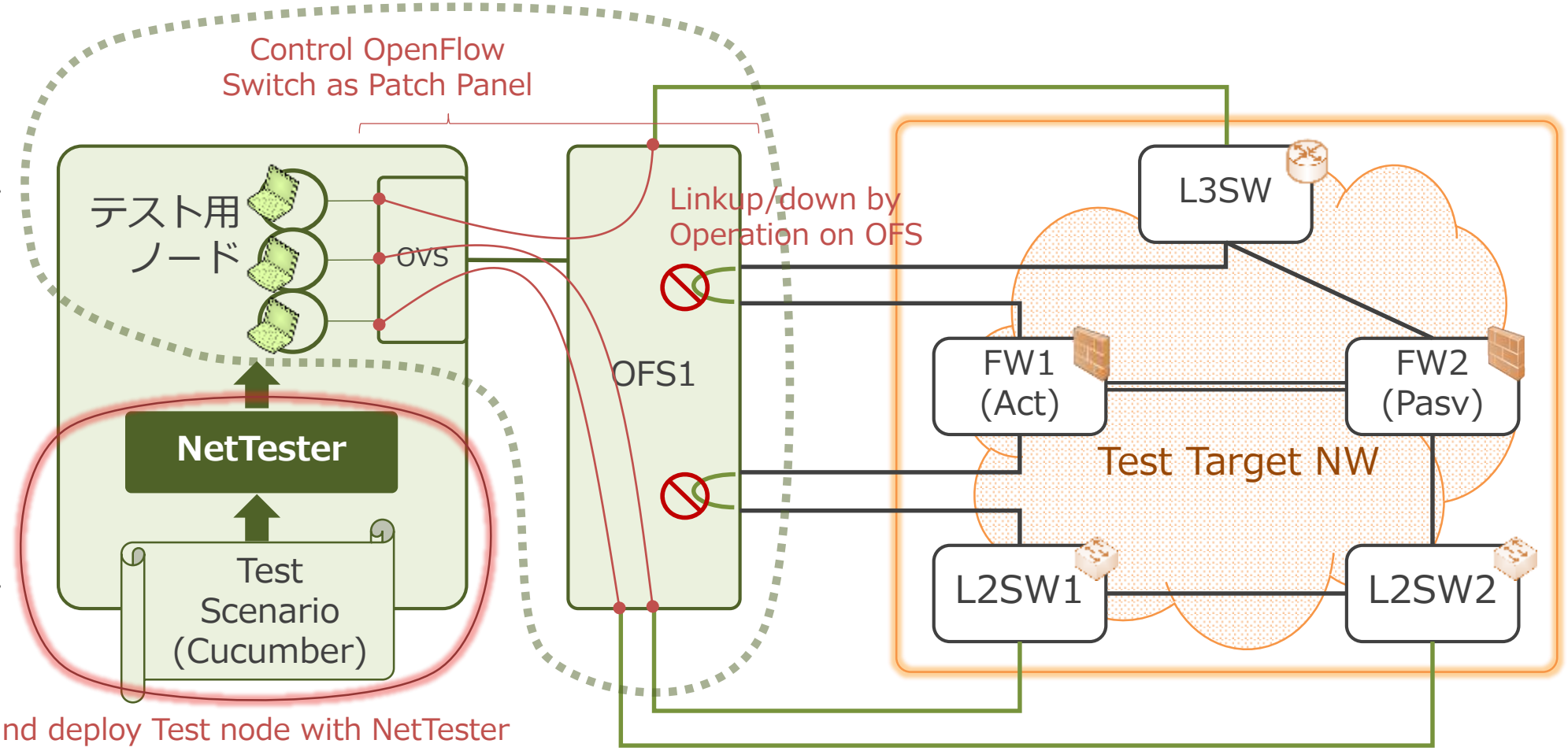
Based on architecture of FY2015

Deploy(Connect) test node to test target

Create and operate NW (Topology) for Test target

Target of FY2016

Generate and operate test node (Execution of each test)



Generate and deploy Test node with NetTester API (Patch panel)
Operate Test node (Test Execution)

Demonstration

Demo movie with explanation

Test automation by NetTester! -Network Test System Project-

https://youtu.be/C7z3aaWgsf4

Overview of Demonstration

Design, Configuration

- Configuration New Network
  - Scope of Work
  - Design, Configuration of Network by Network Engineer

Acceptance Test

- Test by Network Engineer
  - Communication (Functional) Test
  - Switch/Rollback test for failure protection of FW

Change Spec

- Request for Change of IP address of VPN Server
  - Network Configuration change by Network engineer

Acceptance Test

- Re-test by Network Engineer
  - Change Acceptance Test
  - Confirm new Requirement of Network

Design, Configuration

- Configuration New Network
  - Scope of Work
  - Design, Configuration of Network by Network Engineer

Acceptance Test

- Test by Network Engineer
  - Communication (Functional) Test
  - Switch/Rollback test for failure protection of FW

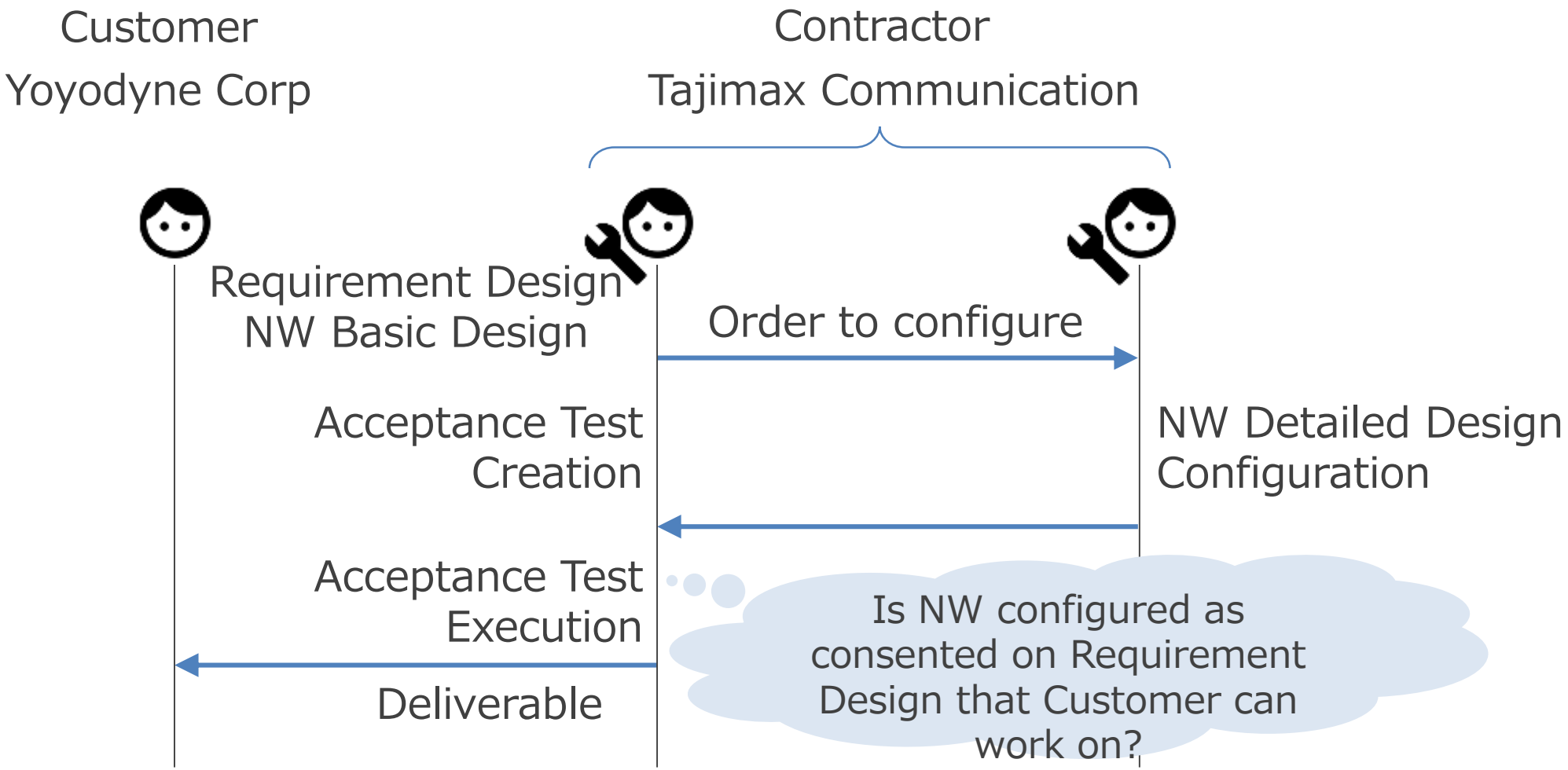
Change Spec

- Request for Change of IP address of VPN Server
  - Network Configuration change by Network engineer

Acceptance Test

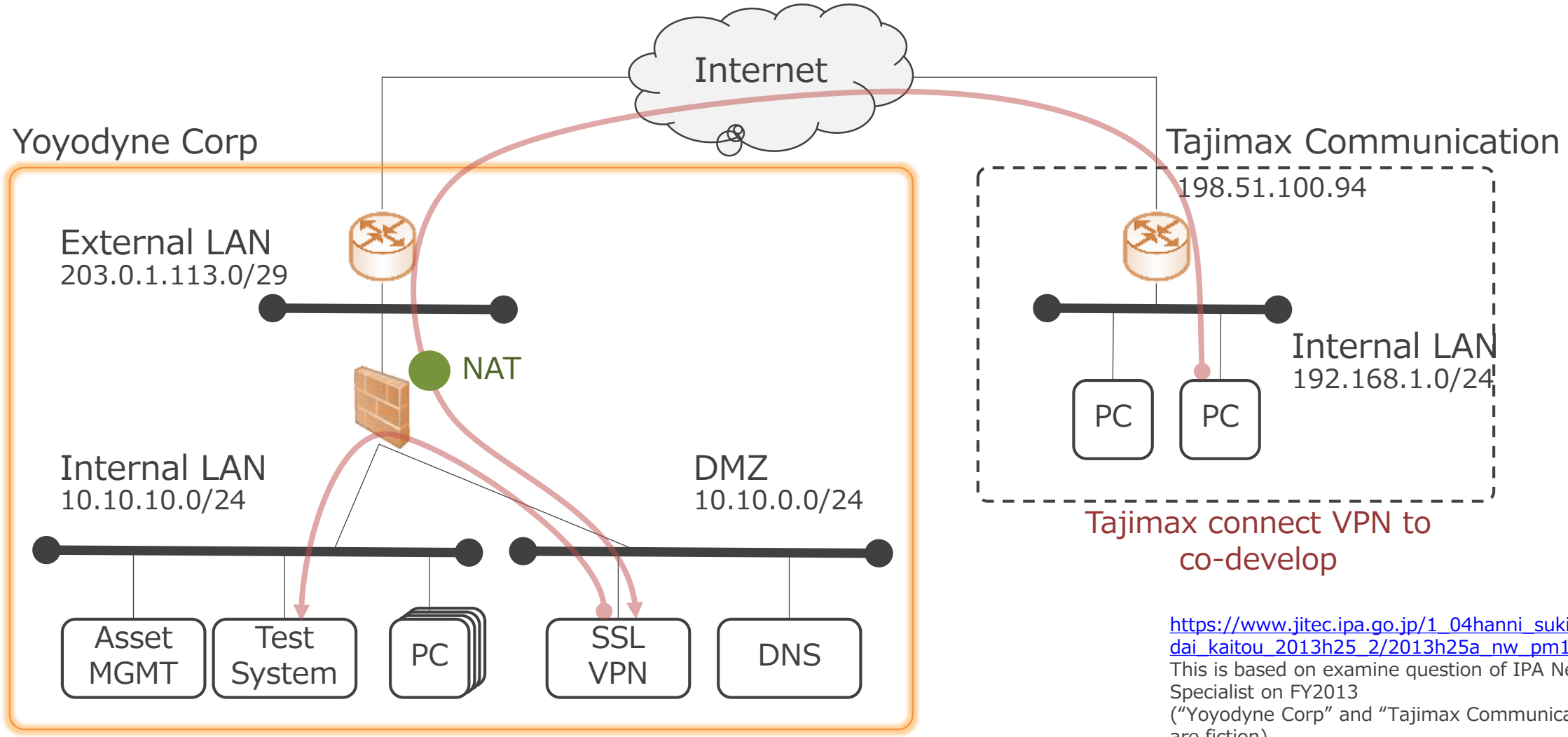
- Re-test by Network Engineer
  - Change Acceptance Test
  - Confirm new Requirement of Network

Story of PoC



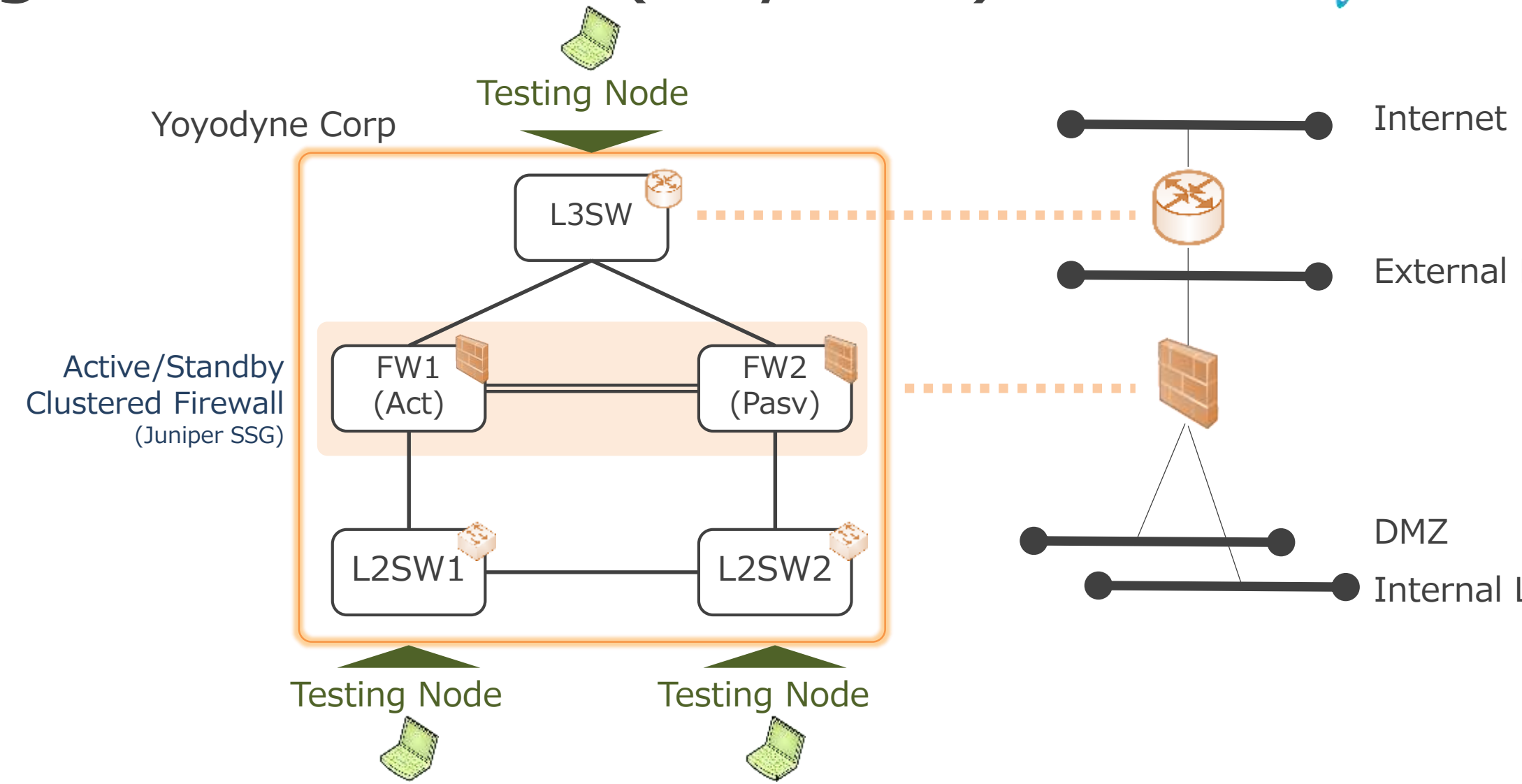
(「Yoyodyne Corp」「Tajimax Communication」は架空の企業名です)

Target of NW Test (Logical)



https://www.jitec.ipa.go.jp/1_04hanni_sukiru/mondai_kaitou_2013h25_2/2013h25a_nw_pm1_qs.pdf
This is based on examine question of IPA Network Specialist on FY2013
("Yoyodyne Corp" and "Tajimax Communication" are fiction)

Target of NW Test (Physical)



Design, Configuration

- Configuration New Network
  - Scope of Work
  - Design, Configuration of Network by Network Engineer

Acceptance Test

- Test by Network Engineer
  - Communication (Functional) Test
  - Switch/Rollback test for failure protection of FW

Change Spec

- Request for Change of IP address of VPN Server
  - Network Configuration change by Network engineer

Acceptance Test

- Re-test by Network Engineer
  - Change Acceptance Test
  - Confirm new Requirement of Network

Example of Test scenario (Dynamic Test)

```
Feature: Stable access for Remote development resources
  Tajimax Communication want to access development resources in Yoyodyne
   Corp as engineer of Tajimax Communication
  Because there is a daily work on remote

  Scenario: Remote connection is kept if linkdown failure occurring
    Given VPN Server in DMZ on Yoyodyne Corp
    And PC of Tajimax Communication as VPN Client
    And operate on server of Yoyodyne Corp from Remote access via VPN
    When Linkdown failure occur between “FW1” and “L2SW1”
    Then Remote access Connection is kept
```

https://github.com/net-tester/examples/blob/feature/ood_demo/features/tcp_fw1_l2sw1_linkdown.feature

Generate and Deploy Testing Node

```
Given(/^VPN Server in DMZ on Yoyodyne Corp$/) do
  @vpn_server = Netns.new(attributes_for(:vpn_server))
end
```

Testing Node

https://github.com/net-tester/examples/blob/feature/ood_demo/features/step_definitions/virtual_host.rb

```
sequence :virtual_port_number, 2

factory :vpn_server, class: NetTester::Netns do
  name 'vpn_server'
  dmz_network
  ip_address '10.10.0.11'
  physical_port_number 9
  mac_address {Faker::Internet.mac_address('00')}
end
```

"patch panel" Configuration (Deployment setting of Testing Node)

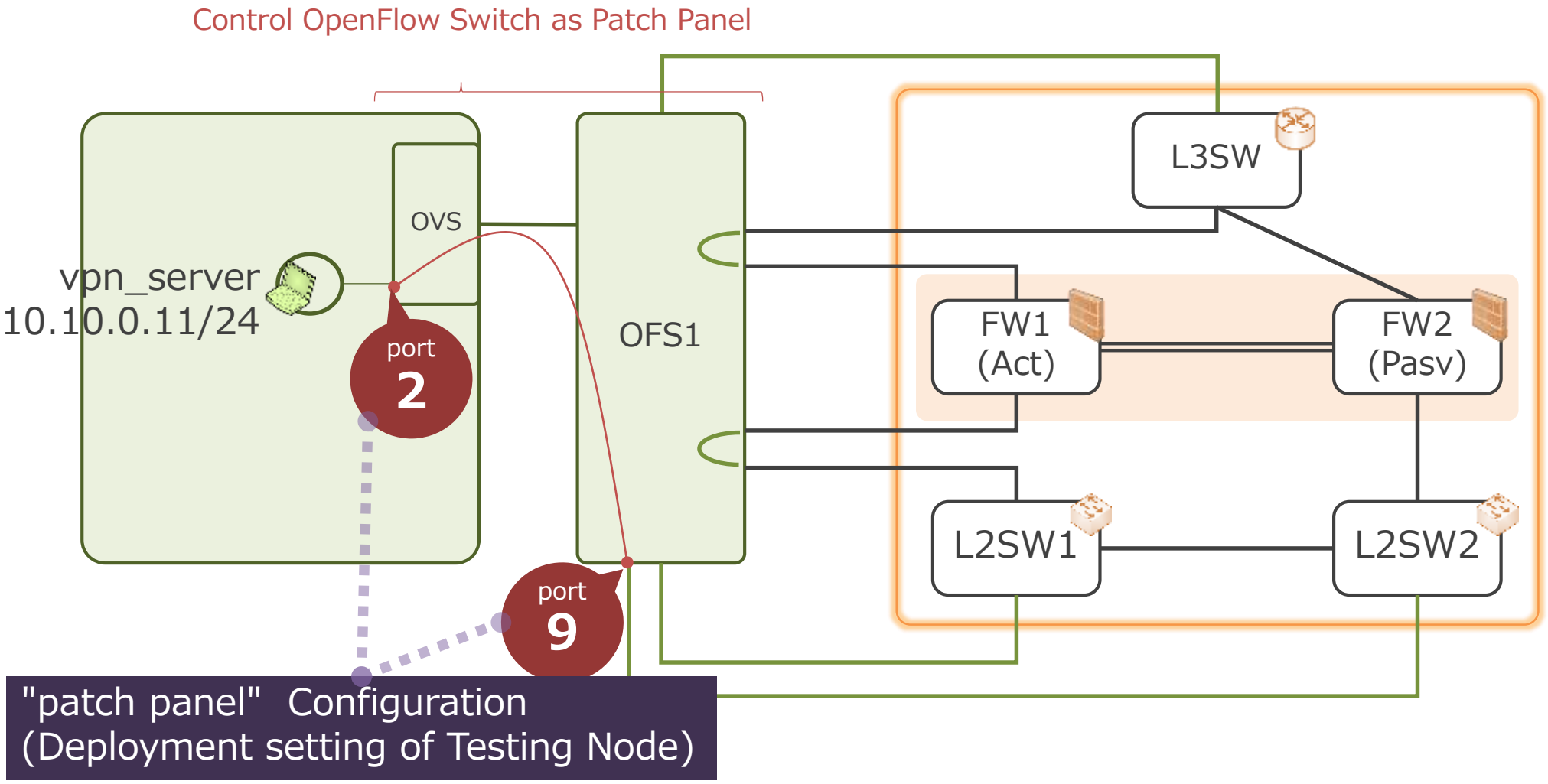
```
trait :dmz_network do
  netmask '255.255.255.0'
  gateway '10.10.0.1'
  virtual_port_number
end
```

Parameter setting of Testing Node

Parameter setting of NW(segment)

https://github.com/net-tester/examples/blob/feature/ood_demo/features/factories.rb

Generate and Deploy Testing Node



Operate Testing Node(Test Execution)

```
Given(/^Operate Operate in Server of Yoyodyne Corp by remote access via VPN$/) do
  step %(Continuous Ping from PC of Tajimax Communication to VPN Server in DMZ on Yoyodyne Corp)
  step %(Initiate TCP session from PC of Tajimax Communication to VPN Server in DMZ on Yoyodyne Corp)
end
```

https://github.com/net-tester/examples/blob/feature/ood_demo/features/step_definitions/remotework_linkdown_steps.rb

```
When(/^ Initiate TCP session from PC of Tajimax Communication to VPN Server in DMZ on Yoyodyne Corp
$/) do
  cd('.') do
    @echo_server = AsyncExecutor.new(host: @vpn_server, result_file: 'log/tcp_server.log')
    @echo_server.exec("../../features/support/echo_server.pl 80")
    @echo_client = AsyncExecutor.new(host: @tajimax_pc, result_file: 'log/tcp_a.log')
    @echo_client.exec("../../features/support/echo_cli                1 203.0.113.5 80 30")
  end
end
```

Get logs by executing tcp echo server/client on Testing Node ("tcp ping")

https://github.com/net-tester/examples/blob/feature/ood_demo/features/step_definitions/continuous_tcp_steps.rb

Operate Testing Node (Judge test result)

```
Then(/^Remote connection is kept$/) do
  step %(Ping from PC of Tajimax Communication to VPN Server in DMZ on Yoyodyne Corp is recovered
within 10 seconds)
  step %(TCP session from PC of Tajimax Communication to VPN Server in DMZ on Yoyodyne Corp is kept
alive)
  step %(FW of Act side is Passive, and Stby side is Active)
end
```

https://github.com/net-tester/examples/blob/feature/ood_demo/features/step_definitions/remotework_linkdown_steps.rb

```
Then(/^ TCP session from PC of Tajimax Communication to VPN Server in DMZ on Yoyodyne Corp is kept
alive $/) do
  @echo_client.join
  cd('.') do
    line_count, _ = check_connection('log/t
    expect(line_count).to be == 3
  end
end
```

Confirm log that there is “no disconnect”

https://github.com/net-tester/examples/blob/feature/ood_demo/features/step_definitions/continuous_tcp_steps.rb

Physical Configuration Change for Target of NW Test

```
When(/^Linkfailure occur between “FW1” and
“L2SW1”$/) do
  step %(Wait 10 seconds)
  step %(Linkfailure occur link of FW1-L2SW1)
end
```

https://github.com/net-tester/examples/blob/feature/ood_demo/features/step_definitions/remotework_linkdown_steps.rb

```
When(/^Linkfailure between link of
FW1-L2SW1$/) do
  cd('.') do
    make_port_down(14)
    make_port_down(15)
  end
end
```

https://github.com/net-tester/examples/blob/feature/ood_demo/features/step_definitions/fw_failure_steps.rb

Execute Port down command after login OpenFlow Switch (Pica8)

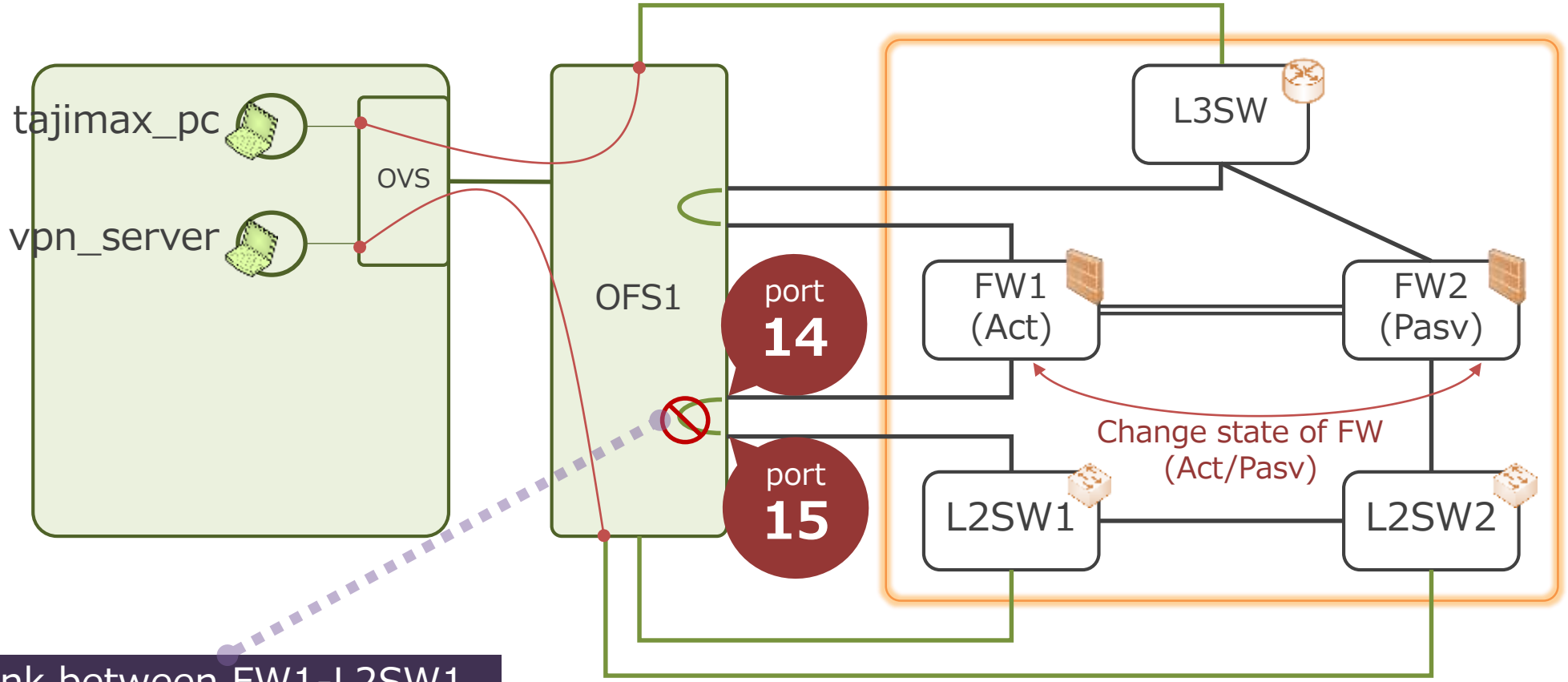
```
def make_port_down(port)
  thrower = Expectacle::Thrower.new(base_dir: __dir__ + '/../support/expectacle', logger: :syslog, verbose: false)
  pica8_hosts = YAML.load_file("#{thrower.hosts_dir}/pica8_hosts.yml")
  pica8_commands = YAML.load_file("#{thrower.commands_dir}/pica8_port_#{port}_down.yml")
  thrower.run_command_for_all_hosts(pica8_hosts, pica8_commands)
end
```

https://github.com/net-tester/examples/blob/feature/ood_demo/features/step_definitions/util.rb

```
- "ovs-ofctl mod-port br0 14 down"
```

https://github.com/net-tester/examples/blob/feature/ood_demo/features/support/expectacle/commands/pica8_port_14_down.yml

Physical Configuration Change for Target of NW Test



Link between FW1-L2SW1
Linkdown/Linkup operation

Design, Configuration

- Configuration New Network
  - Scope of Work
  - Design, Configuration of Network by Network Engineer

Acceptance Test

- Test by Network Engineer
  - Communication (Functional) Test
  - Switch/Rollback test for failure protection of FW

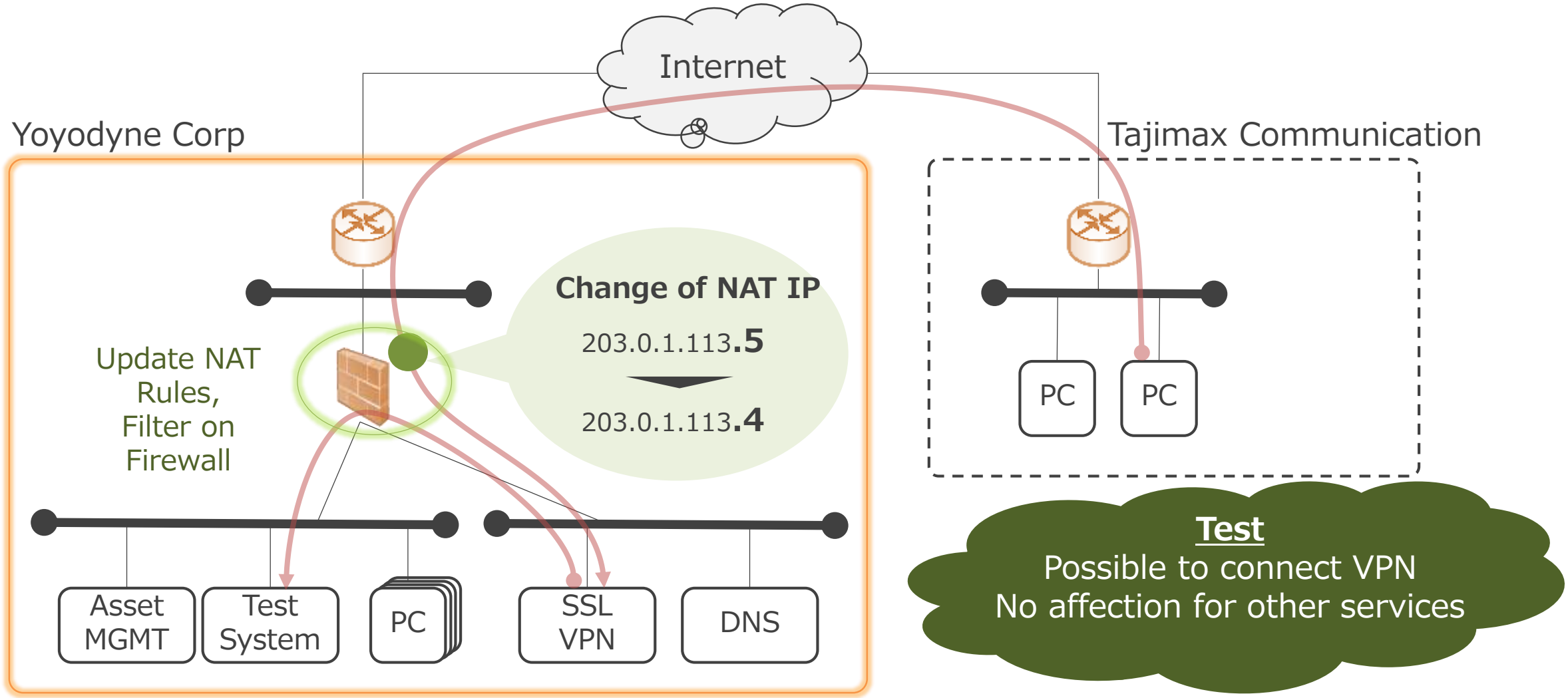
Change Spec

- Request for Change of IP address of VPN Server
  - Network Configuration change by Network engineer

Acceptance Test

- Re-test by Network Engineer
  - Change Acceptance Test
  - Confirm new Requirement of Network

Change Specification



Design, Configuration

- Configuration New Network
  - Scope of Work
  - Design, Configuration of Network by Network Engineer

Acceptance Test

- Test by Network Engineer
  - Communication (Functional) Test
  - Switch/Rollback test for failure protection of FW

Change Spec

- Request for Change of IP address of VPN Server
  - Network Configuration change by Network engineer

Acceptance Test

- Re-test by Network Engineer
  - Change Acceptance Test
  - Confirm new Requirement of Network

Update and Re-execute Test scenario

```
diff --git a/features/step_definitions/continuous_ping_steps.rb b/features/step_definitions/continuous_ping_steps.rb

index 05f6229..1b6860f 100644
--- a/features/step_definitions/continuous_ping_steps.rb
+++ b/features/step_definitions/continuous_ping_steps.rb
@@ -3,7 +3,7 @@
 When(/^Continuous Ping from PC of Tajimax Communication to VPN Server in DMZ on Yoyodyne Corp$/) do
   cd('.') do
     @ping_client = AsyncExecutor.new(host: @tajimax_pc, result_file: 'log/ping_a.log')
-    @ping_client.exec("ping -D -i 0.1 -c 300 203.0.113.5")
+    @ping_client.exec("ping -D -i 0.1 -c 300 203.0.113.4")
   end
 end

diff --git a/features/step_definitions/continuous_tcp_steps.rb b/features/step_definitions/continuous_tcp_steps.rb
index 40726c9..889efe4 100644
--- a/features/step_definitions/continuous_tcp_steps.rb
+++ b/features/step_definitions/continuous_tcp_steps.rb
@@ -6,7 +6,7 @@
 When(/^Initiate TCP session from PC of Tajimax Communication to VPN Server in DMZ on Yoyodyne Corp$/)
   cd('.') do
     @echo_server.exec("../../features/support/echo_server.pl 80")

     @echo_client = AsyncExecutor.new(host: @tajimax_pc, result_file: 'log/tcp_a.log')
-    @echo_client.exec("../../features/support/echo_client.pl 203.0.113.5 80 30")
+    @echo_client.exec("../../features/support/echo_client.pl 203.0.113.4 80 30")
   end
:
```

Modify Acceptance Test according to request for change (Specification)

Result

- **Describe and Execute test includes topology manipulation**
  - ex: Linkdown, failure test
  - Quicker and more precise then Manual operation (Develop as No mistake)
  - NW failure test was executed from Tokyo by remote access to Okinawa
- **Execute Cycle of Modify → Test → Deployment Quickly**
  - Once create test scenario, it could be modified and executed test for changed scenario
- Found Trouble for appliance
  - ex: old SSG → unstable behavior of ARP (Prevented by OS update)
  - It's difficult to be found on virtual appliance
- Configure Practical and complicated functional test
  - ex: Functional test for DPI filter (DNS)

Difficulties…

- Teardown process
  - Repeat test: **Test configuration or status was remained on Physical NW devices**
    - Big difference from application or virtual appliance test
  - Initialization and teardown remained status should be considered

- Cause investigation of “Not executed test”(Trouble shooting and find suspicious factor)
  - Test tools (NetTester+OFS), Configuration of test target, physical connection between devices...

Summary

Summary

OOL developed NW Acceptance Test tool (NetTester) based on FY2015 together with Trema team

- Test “Behavior” of NW
  - Expectation for NW
    =What should be realized on NW?
- Utilizing Knowhow of software development in past
  - Collaborate with BDD tool (Cucumber)
- Realization
  - Not only ”static test”, but also “Dynamic test” executed manually in past can be automated!

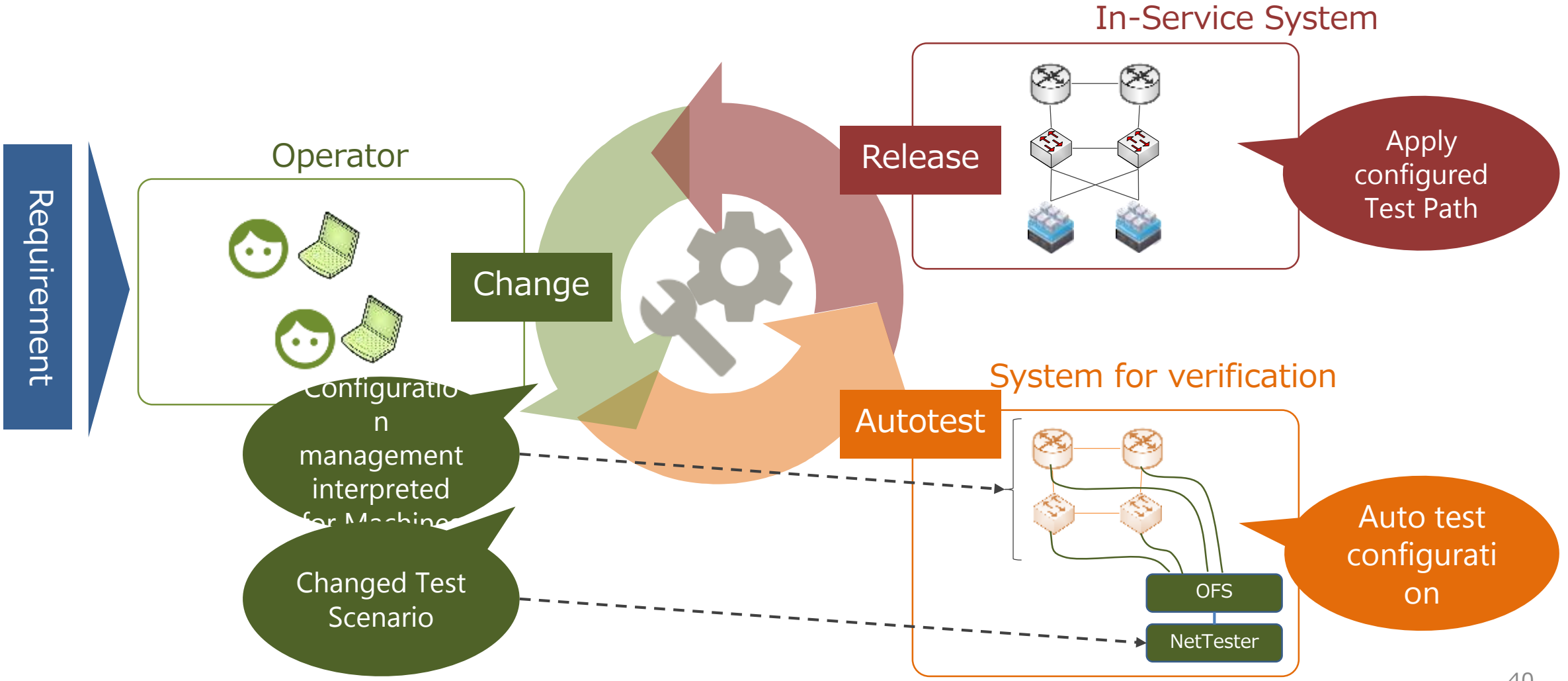
Guarantee value provided by NW service from customer’s eye

Follow changing of requirement for NW more flexibly and quickly

Vision of configuration, operation of infrastructure

OKINAWA OPEN LABORATORY

Continuous test of infrastructure and delivery of service by Software



Further Issues

- Test to find “what is not allowed”
  - ex: Test Filter of FW to judge it is not improper (Open Too many Ports)
  - Other point of view from “Acceptance test for customer use”
- System collaboration to apply process of CI/DevOps
- **Apply to service**: Use in real operation

We are looking for a partner who want to automate NW test together

Reference

NetTester

- net-tester · GitHub
  https://github.com/net-tester
- TestScenario
  net-tester/examples
  https://github.com/net-tester/examples/tree/feature/ood_demo
- Demo movie with explanation
  Test automation with NetTester! -Network Test System Project-
  https://youtu.be/C7z3aaWgsf4
- Demo movie (Screencast)
  https://asciinema.org/a/c9n8xrwxfofpoxvb306ucmb94

OOL Activity of FY2015

- L1Patch applied NW test system | Okinawa Open Laboratory
  http://www.okinawaopenlabs.org/archives/research2014/150410
- Code of PoC FY2015
  GitHub - oolorg/ool-l1patch-dev
  https://github.com/oolorg/ool-l1patch-dev